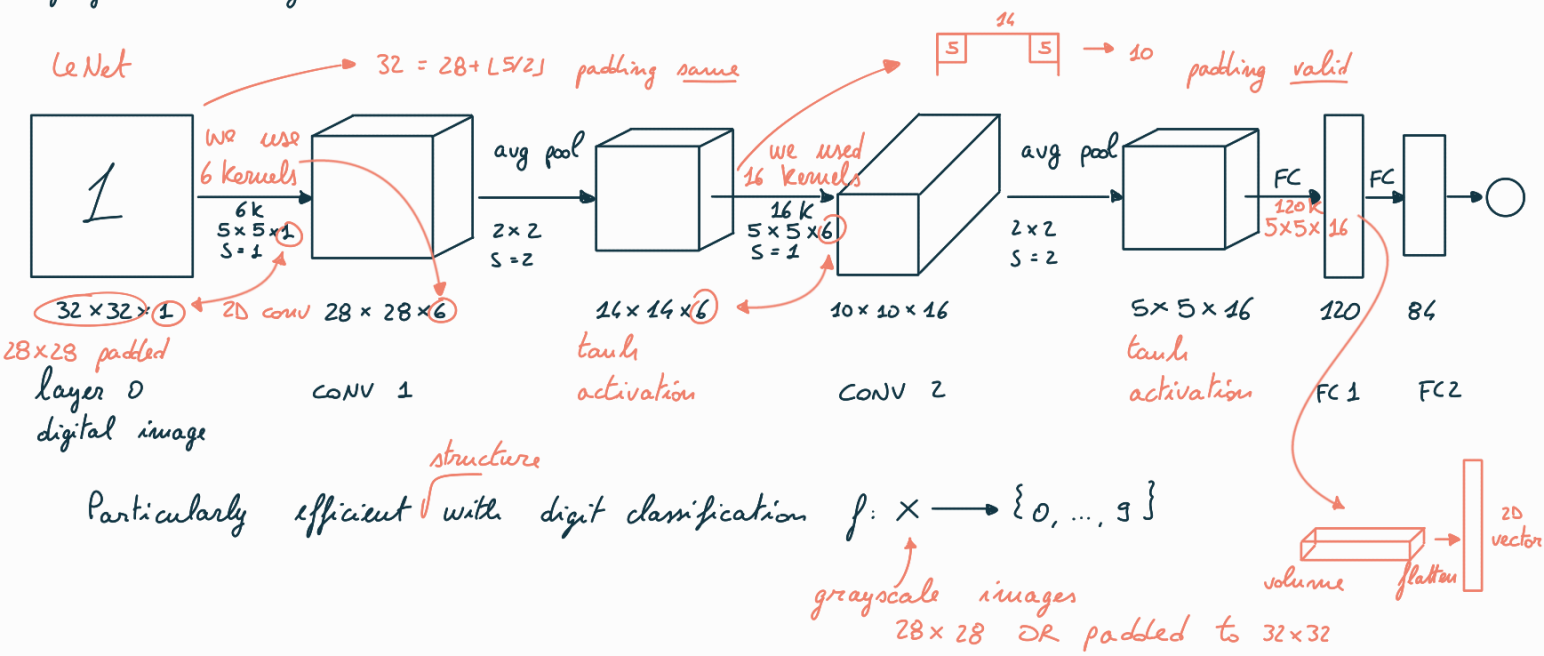
Convolutional Neural Networks for images (2D input)

Every convolutional layer of a CNN transforms the 3D input volume to a 3D output volume of neuron activations. We iterate this process several times and at the end the goal is to reach a situation in which we have a smaller volume. Usually the size of the volume after some levels is much smaller than the size of the input. Once we have this situation we can do what's called a flatten operation, that will produce just a vector of real values; we will continue with a standard fully connected layer.



Particularly efficient (structure) with digit classification $f: X \longrightarrow \{0, \dots, 9\}$

grayscale images 28×28 OR padded to 32×32

2 convolutional layers + 2 fully connected layers

CONV	1	6 k $5 \times 5 \times 1$	$\longrightarrow$	$6 \cdot 5 \cdot 5 + 6$ (bias)
CONV	2	16 k $5 \times 5 \times 6$	$\longrightarrow$	$16 \cdot 5 \cdot 5 \cdot 6 + 16$
CONV	3	120 k $5 \times 5 \times 16$	$\longrightarrow$	$120 \cdot 5 \cdot 5 \cdot 16 + 120$
FC	1	120 i 84 o	$\longrightarrow$	$120 \cdot 84 + 84$
FC	2	84 i 10 o	$\longrightarrow$	$84 \cdot 10 + 10$

(i = input, o = output)

trainable parameters
They are usually few at the beginning but they increase as the network goes deeper.

ILSVRC

2012	AlexNet	(8 layers)
2013	GoogLeNet	(8 layers)
2014	VGG	(19 layers)
2015	ResNet	(22 layers)

AlexNet was the winner of ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2012 (8 layers splitted in two parts trained in parallel to speed up computation - not required with modern hardware)

ResNet was the winner of ILSVRC in 2015. It introduced a new technique: skip connections. With skip connections you skip some of the layers and directly propagate the values some layers further without changes